

Multisensor Out of Sequence Data Fusion for Estimating the State of Discrete Control Systems

Eva Besada-Portas, Jose A. Lopez-Orozco, *Member, IEEE*,
Juan A. Besada, and Jesus M. de la Cruz, *Member, IEEE*

Abstract—The fusion center of a complex control system estimates its state with the information provided by different sensors. Physically distributed sensors, communication networks, pre-processing algorithms, multitasking, etc., introduce non-systematic delays in the arrival of information to the fusion center, making the information available Out-Of-Sequence (OOS). For real-time control systems, the state has to be efficiently estimated with all the information received so far. So, several solutions of the OOS problem for dynamic Multiple-Input Multiple-Output (MIMO) discrete control systems traditionally solved by the Kalman Filter (KF) have been proposed recently. This paper presents two new streamlined algorithms for the linear and non-linear case. IFAsyn, the linear algorithm, is equivalent to other optimal solutions but more general, efficient and easy to implement. EIFAsyn, the nonlinear one, is a new solution of the OOS problem in the Extended KF (EKF) framework.

Index Terms—Discrete control systems, estimation, out-of-sequence (OOS) data.

I. INTRODUCTION

The state of a complex control system is estimated by its fusion center with the information provided by its sensors. The time and order of arrival of the data to the fusion center depends on the localization of the sensors, the time needed to pre-process the measurements, the communication network used to send the data, etc. The most difficult scenario for the fusion center occurs when the delays and the sequence of arrival of all the data are not fixed, constituting the named Out-Of-Sequence Problem (OOSP) [1].

For sequential fusion methods such as the KF [2], the OOSP can be easily tackled discarding the delayed data or buffering all the information relevant to the current state before estimating it. However, those options are not appropriated for networked real-time control systems with many OOS measurements, because they increase the uncertainty of the state and can affect the stability of the feedback loops [3], [4]. Another option consists in 1) storing the estimates of the state, control signals, and sensor data; 2) rolling back to the time stamp associated with the new measurement; and 3) restarting the fusion process from that point. This lets the fusion center obtain the same estimates as if it had received the data without delays, increasing its memory and computational needs. To reduce these needs, several OOS estimators have been recently developed, such as [1] and [5]–[13] for the KF, and [14]–[16] for Particle Filters (PF).

We present two memory and computational efficient centralized algorithms to estimate the state of MIMO discrete control systems with additive non-correlated Gaussian noise in the transition and measurement models with all the already received data with random delays. The first, designed for linear systems and originally proposed in [17], [18], finds the same solutions as the KF or Information Filter (IF) when the measurements are not delayed. It is computationally more efficient, simple to implement and general in scope than [5]–[12]. The second, for nonlinear systems, is a novel approach inside the EKF and Extended IF (EIF), significantly different to the OOS PF [14]–[16] or the use of linear OOS algorithms for linearized systems [19]. It can either find exactly the same solution as the EKF with non delayed measurements or a quicker approximation.

II. ESTIMATING THE STATE OF A MIMO LINEAR CONTROL SYSTEM WITH OOS MEASUREMENTS

A. IFAsyn: An Algorithm for the OOSP Based on the IF

The behavior of a MIMO linear discrete control system with additive white noise and S sensors is modeled by (1), where \mathbf{x}_t is the state of the system, $z_{s,t}$ the measurement of sensor s at time t , $\mathbf{u}_{t,t-1}$ the control signal for that sampling period, $\mathbf{F}_{t,t-1}$ and $\mathbf{H}_{s,t}$ the transition and measurement matrixes, and $\mathbf{v}_{t,t-1}$ and $\mathbf{v}_{s,t}$ gaussian variables with zero mean and covariances $\mathbf{Q}_{t,t-1}$ and $\mathbf{R}_{s,t}$

$$\begin{aligned}\mathbf{x}_t &= \mathbf{F}_{t,t-1}\mathbf{x}_{t-1} + \mathbf{u}_{t,t-1} + \mathbf{v}_{t,t-1} \\ z_{s,t} &= \mathbf{H}_{s,t}\mathbf{x}_t + \mathbf{v}_{s,t} \quad \text{with } s = 1 : S.\end{aligned}\quad (1)$$

Our objective is to estimate the current system state and covariance ($\hat{\mathbf{x}}_{t|t}$, $\mathbf{P}_{t|t}$) given their original values (\mathbf{x}_0 , \mathbf{P}_0), the model parameters and control signals $\{\mathbf{F}_{k,k-1}, \mathbf{Q}_{k,k-1}, \mathbf{H}_{s,k}, \mathbf{R}_{s,k}, \mathbf{u}_{k,k-1} | s = 1 : S, k = 1 : t\}$, and all the data $\{\xi_{s,k,a} = z_{s,k} | s \in 1 : S, k \in 1 : t, a \leq t\}$ measured by sensor s at time k which has already arrived at the fusion center at time t ($a \leq t$). When the data is not delayed ($\xi_{s,k,a} = z_{s,k}$, $a = k$), ($\hat{\mathbf{x}}_{t|t}$, $\mathbf{P}_{t|t}$) can be optimally estimated using the prediction and assimilation steps of the KF or IF.

For assimilating delayed measurements ($\xi_{s,k,a}$, $a > k$), we propose IFAsyn (IF for Asynchronous measurements) that obtains the same results as the KF or IF with the same data without delays. It combines the KF and IF because the KF prediction (2), carried out in the state space ($\hat{\mathbf{x}}_{t|t}$, $\mathbf{P}_{t|t}$), and the IF assimilation for multiple sensors (3), performed in the information space ($\hat{\mathbf{y}}_{t|t}$, $\mathbf{Y}_{t|t}$), are simpler than their counterparts [2]. The IF assimilation is organized as a projection of the measurement into the information space and accumulation of all the projected measurements ((3), (I)), and the assimilation of the accumulated data with the previous information ((3), (A)). This second part naturally separates all the sensorial information previous to t ($\hat{\mathbf{y}}_{t|t-1}$, $\mathbf{Y}_{t|t-1}$) from the sensorial information of instant t ($\hat{\mathbf{i}}_t$, \mathbf{I}_t). Finally, to carry out KF predictions and IF assimilations sequentially, IFAsyn also needs the projection operation between their working spaces $\{\hat{\mathbf{y}}_{j|t} = \mathbf{P}_{j|t}^{-1}\hat{\mathbf{x}}_{j|t}, \mathbf{Y}_{j|t} = \mathbf{P}_{j|t}^{-1}\mathbf{I}_j\} (\perp)$

$$\left. \begin{aligned}\hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_{t,t-1}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{u}_{t,t-1} \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_{t,t-1}\mathbf{P}_{t-1|t-1}\mathbf{F}_{t,t-1}^T + \mathbf{Q}_{t,t-1}\end{aligned}\right\} (P) \quad (2)$$

$$\left. \begin{aligned}\hat{\mathbf{i}}_{s,t} &= \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \xi_{s,t,t}, \mathbf{I}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \\ \hat{\mathbf{i}}_t &= \sum_{s=1}^S \hat{\mathbf{i}}_{s,t}, \mathbf{I}_t = \sum_{s=1}^S \mathbf{I}_{s,t} \\ \hat{\mathbf{y}}_{t|t} &= \hat{\mathbf{y}}_{t|t-1} + \hat{\mathbf{i}}_t, \mathbf{Y}_{t|t} = \mathbf{Y}_{t|t-1} + \mathbf{I}_t\end{aligned}\right\} (I) \quad (3)$$

With all those properties, IFAsyn has a prediction step (4) that is carried out periodically and an assimilation step (5) that is triggered as soon as a new piece of data ($\xi_{s,k,t}$) measured by sensor s at time k is received by the fusion center at time t .

IFAsyn stores for each step time j : $(\hat{\mathbf{y}}_{j|j-1}, \mathbf{Y}_{j|j-1})$ for the information state with all the data previous to j , $(\hat{\mathbf{i}}_j, \mathbf{I}_j)$ for accumulating the sensorial information of instant j already received and projected into the information space, and $(\mathbf{u}_{t,j-1})$ for the control signal applied in each time step. For the current time, it also stores $(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t})$. To reduce the memory needs, IFAsyn only stores the last W group of variables, which lets it only assimilate the measurements that arrive with a maximum delay M_d ($M_d = W - 1$).

The prediction step (4) is the KF prediction (2) plus the initialization (S) of $(\hat{\mathbf{i}}_j, \mathbf{I}_j, \hat{\mathbf{y}}_{j|j-1}, \mathbf{Y}_{j|j-1})$. The assimilation step (5) consists of two different parts. The first (I) projects the new measurement into the information space $(\hat{\mathbf{i}}_{s,k}, \mathbf{I}_{s,k})$ and adds it to $(\hat{\mathbf{i}}_k, \mathbf{I}_k)$. The second is a loop that restarts the assimilation with the already projected sensorial information posterior to k : (A) is the combination of the sensorial information previous to j and the one associated with j ; (P) is the prediction that repropagates all the previous information (including the new measurement) to the next time step $j + 1$; and (\perp) are the projections between the KF and IF spaces.

(4) is equivalent to the KF prediction and (5) restarts the IF from the time stamp associated to the new measurement with the *already projected data* with a bigger timestamp which was received before. So, as the KF prediction, IF assimilation and (\perp) constitute an optimal Minimum Mean Square Error (MMSE) algorithm for non delayed data, IFAsyn is an optimal MMSE algorithm for the OOSP of (1). Its memory and computation benefits are due to the use of projected sensorial measurements

$$\left. \begin{aligned} &\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t,t-1} \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{u}_{t,t-1} \\ &\mathbf{P}_{t|t-1} = \mathbf{F}_{t,t-1} \mathbf{P}_{t-1|t-1} \mathbf{F}_{t,t-1}^T + \mathbf{Q}_{t,t-1} \\ &\left\{ \hat{\mathbf{y}}_{t|t-1} = \mathbf{P}_{t|t-1}^{-1} \hat{\mathbf{x}}_{t|t-1}, \mathbf{Y}_{t|t-1} = \mathbf{P}_{t|t-1}^{-1} \right\} (\perp) \\ &\left\{ \hat{\mathbf{i}}_t = \mathbf{0}, \mathbf{I}_t = \mathbf{0} \right\} \end{aligned} \right\} (P) \quad (4)$$

$$\left. \begin{aligned} &\hat{\mathbf{i}}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \hat{\mathbf{x}}_{s,k,t}, \mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k} \\ &\hat{\mathbf{i}}_k = \hat{\mathbf{i}}_k + \hat{\mathbf{i}}_{s,k}, \mathbf{I}_k = \mathbf{I}_k + \mathbf{I}_{s,k} \\ &\text{for } j = k : t \\ &\quad \left\{ \hat{\mathbf{y}}_{j|j} = \hat{\mathbf{y}}_{j|j-1} + \hat{\mathbf{i}}_j, \mathbf{Y}_{j|j} = \mathbf{Y}_{j|j-1} + \mathbf{I}_j \right\} (A) \\ &\quad \left\{ \hat{\mathbf{x}}_{j|j} = \mathbf{Y}_{j|j}^{-1} \hat{\mathbf{y}}_{j|j}, \mathbf{P}_{j|j} = \mathbf{Y}_{j|j}^{-1} \right\} (\perp) \\ &\text{if } j < t \\ &\quad \left\{ \hat{\mathbf{x}}_{j+1|j} = \mathbf{F}_{j+1,j} \hat{\mathbf{x}}_{j|j} + \mathbf{u}_{j+1,j} \right. \\ &\quad \left. \mathbf{P}_{j+1|j} = \mathbf{F}_{j+1,j} \mathbf{P}_{j|j} \mathbf{F}_{j+1,j}^T + \mathbf{Q}_{j+1,j} \right\} (P) \\ &\quad \left\{ \hat{\mathbf{y}}_{j+1|j} = \mathbf{P}_{j+1|j}^{-1} \hat{\mathbf{x}}_{j+1|j}, \mathbf{Y}_{j+1|j} = \mathbf{P}_{j+1|j}^{-1} \right\} (\perp) \\ &\text{end} \end{aligned} \right\} (5)$$

B. IFAsyn Versus Other Algorithms for the OOSP

This section compares IFAsyn with other OOS algorithms for solving (1) whose optimality holds when the delays are bigger than one time lag ($\xi_{s,t,a}, a > t + 1$) [5]–[11]. We skip [12], optimal for any delay too, because it restarts the fusion process from the time stamp of the incoming $\xi_{s,t,a}$. We also consider [13]-AI, although its degree of suboptimality is a few percent of MSE and is less general because $(\mathbf{F}_{k,l}, \mathbf{Q}_{k,l}, k < l)$ has to be available for arbitrary k and l .

Table I compares the algorithms (column 1) according to their optimality (column 2), the algorithm they are based on (column 3) and their supporting assumptions (columns 4, 5 and 6). Those that invert \mathbf{F} are only valid for discrete systems with non-singular \mathbf{F} (property that holds for all discretized continuous systems). Many doesn't consider $\mathbf{u}_{t,t-1}$, although its inclusion is straightforward in [5], [10], [11]; and more difficult in [7], [8], [13]. Not all of them are designed for multisensor systems: extending [5] is direct; [7], [8], [10] and [11] need considering if there is already a measurement with the same timestamp as the new one, and [13] could be used without changes because it is

TABLE I
IFASYN VERSUS OTHER ALGORITHMS FOR THE OOSP

Algorithm	Optimality	Based	\mathbf{F}^{-1}	\mathbf{u}	Sensors
IFAsyn	v	IF		v	S
[5]	v	KF			1
[6]	v	KF		v	S
[7]-AI	v	KF			1
[8]-AI	v	KF			1
[9]	v	IF	v	v	S
[10]-AI	v	KF			1
[11]	v	KF			1
[13]-AI		KF	v		1

TABLE II
MEMORY COMPARISON OF IFASYN AND OTHER LINEAR ALGORITHMS

Alg.	Memory needs for the last W time steps (state and control signal with n_x elements)	Comparison (best less memory)
IFAsyn	$W(2n_x + 2n_x^2) + Wn_x$	2^{nd}
[5]	$Wn_x + W^2n_x^2$	5^{th}
[6]	$Wn_x + W^2n_x^2$	5^{th}
[7]-AI	$W(2n_x + 4n_x^2) + Wn_x$	4^{th}
[8]-AI	$W(2n_x + 3n_x^2) + Wn_x$	3^{rd}
[9]	$W(n_x + n_x^2) + Wn_x$	1^{st}
[10]-AI	$W(n_x + n_x^2) + Wn_x$	1^{st}
[11]	$W(n_x + n_x^2) + Wn_x$	1^{st}
[13]-AI	$W(n_x + n_x^2) + Wn_x$	1^{st}

already an approximated algorithm. Finally, the simplicity of IFAsyn, only exceeded by [5], and its capacity to deal with multiple sensors comes from its IF support. [9] also shares those properties, but it is less general because it inverts $\mathbf{F}_{t,t-1}$.

To make a fair memory and computational comparison of the algorithms of Table I, we have modified [5]–[11] and [13] to include $\mathbf{u}_{t,t-1}$ and use multiple sensors. Besides, all of them work with the same basic time step and with a window with W time steps. For [13] we also assume that arbitrary $(\mathbf{F}_{k,l}, \mathbf{Q}_{k,l}, k < l)$ are available.

Table II shows their memory needs (column 2), when \mathbf{x}_t and $\mathbf{u}_{t,t-1}$ have n_x elements. It also compares them (column 3) when $W > 2$ (for delays bigger than one time lag). IFAsyn needs more memory than [9]–[11] and [13] because IFAsyn stores $(\hat{\mathbf{y}}_{j|j-1}, \mathbf{Y}_{j|j-1}, \hat{\mathbf{i}}_j, \mathbf{I}_j)$ while [9] stores $(\hat{\mathbf{y}}_{j|j}, \mathbf{Y}_{j|j})$, and [10], [11] and [13] store $(\hat{\mathbf{x}}_{j|j}, \mathbf{P}_{j|j})$. However, the extra memory of IFAsyn makes it more accurate.

To compare the computational cost of each algorithm, we have computed analytically the number of Floating Point Operations (FLOPs) of a prediction step and assimilation step of a $\xi_{s,k,t}$ delayed r time steps ($r = t - k$). IFAsyn, [9]–[11] and [13] have a linear dependency on r , while [5]–[8] depend on W ([5] $\sim W^3$, [6], [7], [8] $\sim W^2$). The asymptotic behavior of their number of FLOPs is shown in Figs. 1, 2 and 3 when we increment the r for a fixed W , the maximum delay ($r = W - 1$), and n_x at $r = W - 1$. As the changing parameter grows, the most efficient algorithms are IFAsyn, [6], [11] and [13]. [13], which is approximated and not valid for any discrete system, is worst than the others for small delays and better when they grow (Figs. 1 and 2). IFAsyn is always better than [11]. Although [6] can outperform IFAsyn (Fig. 3), its number of FLOPs depends on W instead of r (Fig. 1), and it degrades quicker than IFAsyn as the maximum permitted delays grows (Fig. 2). So, for systems which can have random big delays, we consider IFAsyn computational better than [6]. Additionally, IFAsyn is easier to implement.

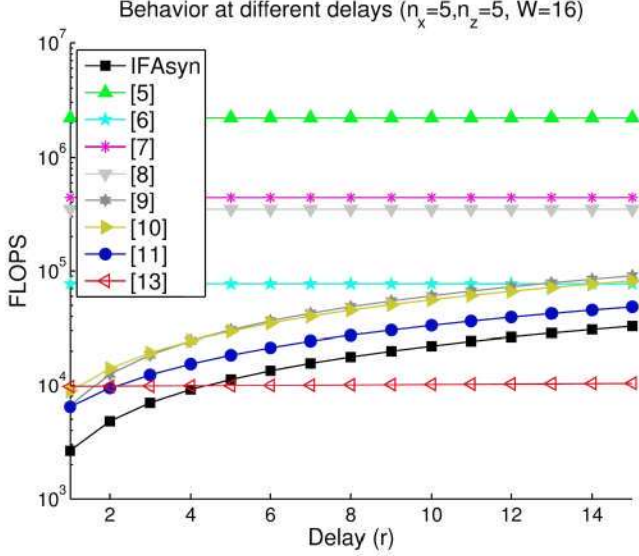


Fig. 1. FLOPs comparison of the linear algorithms as the delay of the measurement grows with a fixed window size.

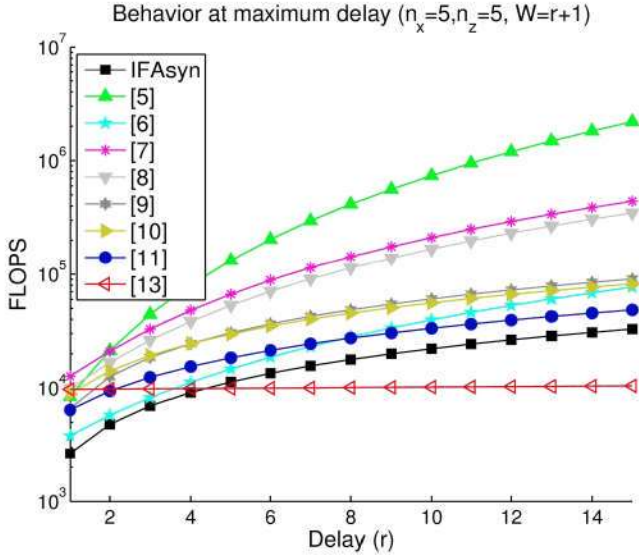


Fig. 2. FLOPs comparison of the linear algorithms as the delay of the measurement, fixed to the maximum permitted one, grows.

The memory and computation comparison show that the best *optimal* algorithms are [11] and IFAsyn. [11] minimizes the memory while only incrementing the computational cost of IFAsyn and [6]. IFAsyn usually minimizes the computation needs of the rest at the expenses of almost doubling the memory of [11]. That is, IFAsyn extra memory is paying for a reduction of the computational needs.

Finally, an analysis of the stability of a control system that uses IFAsyn (or [5], [7]–[11]) is out of the scope of this paper, but its equivalency with [6] makes it share its stability properties. Although they use the OOS data, the magnitude of the delays is important because while the data is not available the control system is working with intermittent observations what can instantaneously affect its stability [3], [4]. The problem is not posed by IFAsyn and [5]–[11], as they update the state as quickly as its computational requirements let them do it, but it has to be considered when they are used inside real-time control loops.

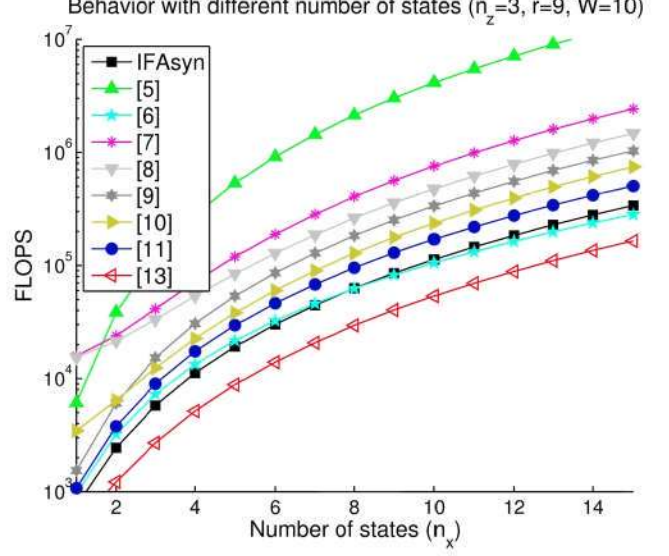


Fig. 3. FLOPs comparison of the linear algorithms as the number of states grows.

III. ESTIMATING THE STATE OF A MIMO NON-LINEAR CONTROL SYSTEM WITH OOS MEASUREMENTS

A. EIFAsyn: An Algorithm for OOSP Based on the EIF

The behavior of a MIMO nonlinear discrete control system with additive white noise and S sensors is modeled with (6), where $(\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{u}_{t,t-1}, \mathbf{v}_{t,t-1}, \mathbf{Q}_{t,t-1}, \mathbf{R}_{s,t})$ have the same meaning as in Section II-A, and $f(\cdot)$ and $h_s(\cdot)$ are the transition and measurements functions. They can be linear too, and so the general problem can be completely linear, completely nonlinear, or a mixture of both

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}, \mathbf{u}_{t,t-1}, t, t-1) + \mathbf{v}_{t,t-1} \\ \mathbf{z}_{s,t} &= h_s(\mathbf{x}_t, t) + \mathbf{v}_{s,t} \quad \text{with } s = 1 : S. \end{aligned} \quad (6)$$

The objective of the fusion center is the same as in the linear case, but with the new models $f(\cdot)$ and $h_s(\cdot)$. When the data arrives without delays $(\xi_{s,k,a}, a = k)$, $(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t})$ can be approximately obtained by the prediction and assimilation steps of EKF and EIF [2]. The KF prediction and IF assimilation are presented in (7) and (8), where $(\mathbf{J}g)_x(\cdot)$ is the Jacobian of function $g(\cdot)$

$$\left. \begin{aligned} \mathbf{F}_{t,t-1} &= (\mathbf{J}f)_x(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_{t,t-1}, t, t-1) \\ \hat{\mathbf{x}}_{t|t-1} &= f(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_{t,t-1}, t, t-1) \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_{t,t-1} \mathbf{P}_{t-1|t-1} \mathbf{F}_{t,t-1}^T + \mathbf{Q}_{t,t-1} \end{aligned} \right\} (P) \quad (7)$$

$$\left. \begin{aligned} \mathbf{H}_{s,t} &= (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t|t-1}, t) \\ \xi_{s,t,t}^C &= \xi_{s,t,t} + (\mathbf{H}_{s,t} \hat{\mathbf{x}}_{t|t-1} - h_s(\hat{\mathbf{x}}_{t|t-1}, t)) \\ \mathbf{i}_{s,t} &= \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \xi_{s,t,t}^C; \mathbf{I}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \\ \mathbf{i}_t &= \sum_{s=1}^S \mathbf{i}_{s,t}; \mathbf{I}_t = \sum_{s=1}^S \mathbf{I}_{s,t} \\ \{\hat{\mathbf{y}}_t\} &= \{\hat{\mathbf{y}}_{t|t-1} + \mathbf{i}_t, \mathbf{Y}_{t|t} = \mathbf{Y}_{t|t-1} + \mathbf{I}_t\} (A) \end{aligned} \right\} (I) \quad (8)$$

To assimilate delayed measurements $(\xi_{s,k,a}, a > k)$ we propose EIFAsyn, an adaptation of IFAsyn which is based on the differences which exist between the expressions (2) and (7), and (3) and (8).

The only significant change of the EKF prediction (7) respect the KF prediction (2) is a new operation to obtain $\mathbf{F}_{t,t-1}$. Besides, the predictions (P) that appear in IFAsyn ((4) and (5)) use the whole KF prediction (2). So, substituting in IFAsyn the KF prediction by the EKF

TABLE III
NON LINEAR SYSTEM (T) WITH 3 SENSORS (S1, S2 AND S3)

T	$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + uD_{t+1,t} \cos(\theta_t + uG_{t+1,t}) \\ y_t + uD_{t+1,t} \sin(\theta_t + uG_{t+1,t}) \\ \theta_t + uG_{t+1,t} \end{pmatrix}$
S1	$z_{1,t+1} = \theta_{t+1}$
S2	$z_{2,t+1} = \begin{pmatrix} \sin(2x_{t+1}) & \sin(2y_{t+1}) & \theta_{t+1} \end{pmatrix}^T$
S3	$z_{3,t+1} = \sin^3(2x_{t+1}) + \sin^3(2y_{t+1})$

prediction makes EIFAsyn suitable for general systems with nonlinear transition models and linear measurements.

The discrepancies of the assimilation steps of IF (3) and EIF (8) are more important. The EIF calculates $\mathbf{H}_{s,t}$ and projects a corrected measurement ($\xi_{s,t,t}^C$), that considers the discrepancy between the linearized measurement model ($\mathbf{H}_{s,t}\hat{\mathbf{x}}_{t|t-1}$) and the nonlinear one ($h_s(\hat{\mathbf{x}}_{t|t-1}, t)$), into the information space. So, ($\mathbf{i}_{s,t}$, $\mathbf{I}_{s,t}$) depend on $\hat{\mathbf{x}}_{t|t-1}$. However, the IFAsyn assimilation (5) projects and accumulates the measurements outside its loop because those operations don't depend on $\hat{\mathbf{x}}_{t|t-1}$. So, if for EIFAsyn we maintain them outside the loop, including previously the calculus of $\mathbf{H}_{s,t}$ and ($\xi_{s,t,t}^C$), we won't obtain the same values as the EIF when the measurements are not delayed. To make EIFAsyn behave as EIF, the whole EIF assimilation has to be included inside the loop, losing the memory and computational benefits of IFAsyn. However, the EKF and EIF are already approximated solutions of the nonlinear problem. So, EIFAsyn assimilation supports both options. The first approximates the estimates reducing the computational and memory cost related with some sensors. The second recalculates the projected measurements for the sensors where ($\mathbf{H}_{s,t}\hat{\mathbf{x}}_{t|t-1} - h_s(\hat{\mathbf{x}}_{t|t-1}, t)$) and $\mathbf{H}_{s,t}$ changes significantly when $\hat{\mathbf{x}}_{t|t-1}$ is updated.

The EIFAsyn prediction and assimilation steps are presented in (9) and (10). The variable **rec** defines the set of nonlinear sensors whose values need to be recalculated. Obviously, the linear sensors don't belong to **rec**. Besides, EIFAsyn stores the IFAsyn variables ($\hat{\mathbf{y}}_{j|j-1}$, $\mathbf{Y}_{j|j-1}$, \mathbf{i}_j , \mathbf{I}_j , $\mathbf{u}_{j,j-1}$) and all received $\xi_{s,k,a}$ of the sensors belonging to **rec**. (\mathbf{i}_j , \mathbf{I}_j) are used to accumulate the projected information of the sensors not belonging to **rec**, while (\mathbf{i}_j^{rec} , \mathbf{I}_j^{rec}) are used for the remaining ones. However, (\mathbf{i}_j^{rec} , \mathbf{I}_j^{rec}) are optional. They only make EIFAsyn avoid to recalculate the projected measurement for the time-stamp k of the $\xi_{s,k,t}$ which triggered the assimilation step. For the linear sensors EIFAsyn doesn't calculate ($\mathbf{J}h_s)(\cdot)$ or $\xi_{s,k,t}^C$, and when the transition model is linear ($\mathbf{J}f)(\cdot)$ is not used.

If all the sensors belong to **rec**, EIFAsyn has to recalculate everything from scratch from the time stamp of each new measurement. Therefore, the memory and computational benefits of EIFAsyn depend on the quantity of sensors not belonging to **rec**. So, they should be carefully determined, considering which nonlinear sensors will be affected by older measurements and how small changes in $\hat{\mathbf{x}}_{j+1|j}$ affect $\xi_{s,j+1,a}^C$ and $\mathbf{H}_{s,j+1}$. This analysis is extremely important, because when a sensor which must belong to **rec** doesn't, EIFAsyn can return erroneous estimates (see example in Section III-B).

Finally, EIFAsyn is a suboptimal solution to the OOSP due to its EKF and EIF origin, and in the case of not recalculating the measurements of all the nonlinear sensors, it doesn't obtain the same solutions as the EKF with non-delayed measurements

$$\left. \begin{aligned} &\mathbf{F}_{t,t-1} = (\mathbf{J}f)(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_{t,t-1}, t, t-1) \\ &\hat{\mathbf{x}}_{t|t-1} = f(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_{t,t-1}, t, t-1) \\ &\mathbf{P}_{t|t-1} = \mathbf{F}_{t,t-1}\mathbf{P}_{t-1|t-1}\mathbf{F}_{t,t-1}^T + \mathbf{Q}_{t,t-1} \\ &\left\{ \begin{aligned} &\hat{\mathbf{y}}_{t|t-1} = \mathbf{P}_{t|t-1}^{-1}\hat{\mathbf{x}}_{t|t-1}, \mathbf{Y}_{t|t-1} = \mathbf{P}_{t|t-1}^{-1} \end{aligned} \right\} (\perp) \\ &\left\{ \begin{aligned} &\mathbf{i}_t = \mathbf{0}, \mathbf{I}_t = \mathbf{0}, \mathbf{i}_t^{rec} = \mathbf{0}, \mathbf{I}_t^{rec} = \mathbf{0} \end{aligned} \right\} (S) \end{aligned} \right\} (9)$$

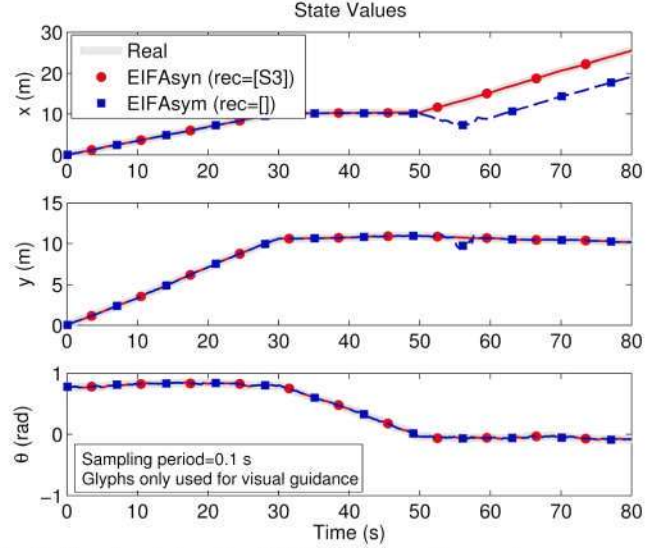


Fig. 4. Experimental results of EIFAsyn for a non linear system.

$$\left. \begin{aligned} &\left\{ \begin{aligned} &\hat{\mathbf{x}}_{k|k-1} = \mathbf{Y}_{k|k-1}^{-1}\hat{\mathbf{y}}_{k|k-1} \\ &\mathbf{H}_{s,k} = (\mathbf{J}h_s)(\hat{\mathbf{x}}_{k|k-1}, k) \\ &\xi_{s,k,t}^C = \xi_{s,k,t} + \mathbf{H}_{s,k}\hat{\mathbf{x}}_{k|k-1} - h_s(\hat{\mathbf{x}}_{k|k-1}, k) \\ &\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \xi_{s,k,t}^C, \mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k} \\ &\text{if } (s \notin \text{rec}) \\ &|\mathbf{i}_k = \mathbf{i}_k + \mathbf{i}_{s,k}, \mathbf{I}_k = \mathbf{I}_k + \mathbf{I}_{s,k} \\ &\text{else} \\ &|\mathbf{i}_k^{rec} = \mathbf{i}_k^{rec} + \mathbf{i}_{s,k}, \mathbf{I}_k^{rec} = \mathbf{I}_k^{rec} + \mathbf{I}_{s,k} \\ &\text{store } \xi_{s,k,t} \end{aligned} \right\} (I) \\ &\text{end} \\ &\text{for } j = k : t \\ &|\mathbf{y}_{j|j} = \hat{\mathbf{y}}_{j|j-1} + \mathbf{i}_j + \mathbf{i}_j^{rec} \\ &|\mathbf{Y}_{j|j} = \mathbf{Y}_{j|j-1} + \mathbf{I}_j + \mathbf{I}_j^{rec} \\ &|\mathbf{y}_{j|j} = \mathbf{Y}_{j|j}^{-1}\hat{\mathbf{y}}_{j|j}, \mathbf{P}_{j|j} = \mathbf{Y}_{j|j}^{-1} \\ &|\text{if } j < t \\ &|\mathbf{F}_{j+1,j} = (\mathbf{J}f)(\hat{\mathbf{x}}_{j|j}, \mathbf{u}_{j+1,j}, j+1, j) \\ &|\hat{\mathbf{x}}_{j+1|j} = f(\hat{\mathbf{x}}_{j|j}, \mathbf{u}_{j+1,j}, j+1, j) \\ &|\mathbf{P}_{j+1|j} = \mathbf{F}_{j+1,j}\mathbf{P}_{j|j}\mathbf{F}_{j+1,j}^T + \mathbf{Q}_{j+1,j} \\ &|\left\{ \begin{aligned} &\hat{\mathbf{y}}_{j+1|j} = \mathbf{P}_{j+1|j}^{-1}\hat{\mathbf{x}}_{j+1|j}, \mathbf{Y}_{j+1|j} = \mathbf{P}_{j+1|j}^{-1} \end{aligned} \right\} (\perp) \\ &|\mathbf{i}_{j+1}^{rec} = \mathbf{0}, \mathbf{I}_{j+1}^{rec} = \mathbf{0} \\ &|\text{for } s \in \text{rec} \\ &|\mathbf{H}_{s,j+1} = (\mathbf{J}h_s)(\hat{\mathbf{x}}_{j+1|j}, j+1) \\ &|\mathbf{c}_{s,j+1} = \mathbf{H}_{s,j+1}\hat{\mathbf{x}}_{j+1|j} - h_s(\hat{\mathbf{x}}_{j+1|j}, j+1) \\ &|\xi_{s,j+1,a}^C = \xi_{s,j+1,a} + \mathbf{c}_{s,j+1} \\ &|\mathbf{i}_{j+1}^{rec} = \mathbf{i}_{j+1}^{rec} + \mathbf{H}_{s,j+1}^T \mathbf{R}_{s,j+1}^{-1} \xi_{s,j+1,a}^C \\ &|\mathbf{I}_{j+1}^{rec} = \mathbf{I}_{j+1}^{rec} + \mathbf{H}_{s,j+1}^T \mathbf{R}_{s,j+1}^{-1} \mathbf{H}_{s,j+1} \\ &|\text{end} \\ &|\text{end} \\ &\text{end} \end{aligned} \right\} (10)$$

B. Example

EIFAsyn is tested with the system of Table III to show the importance of selecting which sensors belong to **rec**. The system has a non-linear transition (T) model, and one linear (S1) and two non-linear (S2, S3) sensors. All the sensors measure every 0.1 second and the measurements of S2 arrive delayed to the fusion center 10 time steps. Fig. 4 shows an experiment where the non-linearities make necessary to recalculate the information of S3. The real state of the system (**Real**) and the state estimated by EIFAsyn re-projecting S3 measurements (**rec = [S3]**) coincide, while the state estimated without the re-projection (**rec = []**) differs. However, other tests with the same system

don't show those differences. So, the properties of each system must be carefully analyzed, paying special attention to how $(\mathbf{H}_{s,j+1}\hat{\mathbf{x}}_{j+1|j} - h_s(\hat{\mathbf{x}}_{j+1|j}, j+1))$ is affected by $\hat{\mathbf{x}}_{j+1|j}$.

C. EIFAsyn versus Other Algorithms for the OOSP

The extension of IFAsyn for the nonlinear transition model is direct, and without a substantial computational overload, it lets EIFAsyn obtain the same solution as the EKF for systems with non-linear transitions and linear measurements. However, many linear algorithms can't be extended for that case: [5], [6] depend on the linearity of the transition which won't hold true, and [7], [8] pre-calculate variables which won't be useful due to the non-linearities.

In contrast, the linear algorithms are sometimes used in systems with linearized non linear models [19]. This can be done when $(\mathbf{H}_{s,j+1}\hat{\mathbf{x}}_{j+1|j} - h_s(\hat{\mathbf{x}}_{j+1|j}, j+1))$ doesn't change significantly, but in other case, the nonlinear measurements need to be re-projected.

The EIF support of EIFAsyn make it not directly comparable with the OOS PF [14]–[16].

Finally, the stability problem of a control system that includes EIFAsyn for estimating the state will be caused by the EKF properties and the magnitude of the delays.

IV. CONCLUSION

We have presented two new algorithms to estimate the state of dynamic control systems for the OOSP. IFAsyn is optimal for linear systems and avoids to recalculate the information of the already received measurements, although it re-propagates it. It is computational better than other optimal solutions and easier to implement as it consists in a cleverly organized group of KF prediction and IF assimilation operations. Besides, it is more general: it doesn't invert the transition matrix, includes the control signal and works with multiple sensors. EIFAsyn is a suboptimal solution for nonlinear systems, equivalent to the EIF when it re-projects the measurements of all the nonlinear sensors and an approximation when it doesn't. The non-linearities in the transitions don't penalize significantly their number of operations and memory needs, while the non-linearities in the sensors can do it.

REFERENCES

- [1] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: Exact solution," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 769–778, Jul. 2002.
- [2] G. O. Mutambara, *Decentralised Estimation and Control for Multi-sensor Fusion*. Boca Raton, FL: CRC, 1998.
- [3] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.
- [4] J. P. Hespanha, P. Naghtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.
- [5] S. Challa, R. J. Evans, X. Wang, and J. Leggy, "A fixed lag-smoothing solution to out-of-sequence information fusion problems," *Commun. Inform. Syst.*, vol. 2, no. 4, pp. 325–348, 2002.
- [6] A. Matveev and A. V. Savkin, "The problem of state estimation via asynchronous communication channels with irregular transmission times," *IEEE Trans. Automat. Control*, vol. 48, no. 4, pp. 670–676, Apr. 2003.
- [7] K. Zhang, X. R. Li, and Y. Chen, "Optimal update with out of sequence measurements for distributed filtering," in *Proc. 5th Int. Conf. Inform. Fusion*, 2002, pp. 1519–1526.
- [8] K. Zhang, X. Li, and Y. Zhu, "Optimal update with out-of-sequence measurements," *IEEE Trans. Signal Processing*, vol. 53, no. 6, pp. 1992–2004, Jun. 2005.
- [9] E. W. Nettleton and H. Durrant-Whyte, "Delayed and asequent data in decentralised sensing networks," in *Proc. SPIE Conf.*, 2001, pp. 1–9.
- [10] K. Zhang, X. Li, and Y. Chen, "Multi-sensor multi-tracking with out-of-sequence measurements," in *Proc. ISIF'03*, 2003, pp. 672–679.
- [11] Z. WenHui, L. Lin, C. GuoHai, and Y. AnXi, "Optimal updater with multistep out of sequence measurements in target tracking," in *Proc. 8th Int. Conf. Signal Processing*, 2006, vol. 4, pp. 281–284.
- [12] H. Zhang, L. Xie, D. Zhang, and Y. C. Soh, "A reorganized innovation approach to linear estimation," *IEEE Trans. Automat. Control*, vol. 49, no. 10, pp. 1810–1814, Oct. 2004.
- [13] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn, "One-step solution for the general out-of-sequence measurement problem in tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 1, pp. 27–37, Jan. 2004.
- [14] M. Orton and A. Marrs, "A bayesian approach to multi-target tracking and data fusion with out-of-sequence measurements," in *Proc. IEEE Int. Workshop Target Tracking Algorithms Appl.*, 2001, pp. 1–15.
- [15] M. Mallick, M. T. Kirubarajan, and S. Arulampalam, "Out-of-sequence measurement processing for tracking ground target using particle filters," in *Proc. IEEE Aerosp. Conf.*, 2002, vol. 4, pp. 1809–1818.
- [16] F. Xue, Z. Liu, and Z. Shi, "Unscented particle filter for bearings-only tracking with out-of-sequence measurements in sensor networks," in *Proc. 6th Int. Conf. Intell. Syst. Design Appl.*, Oct. 2006, vol. 2, pp. 540–545. IEEE.
- [17] E. Besada-Portas, "Fusion multisensorial de medidas temporalmente desordenadas. aplicación a robots autónomos móviles," Ph.D. dissertation, Universidad Complutense de Madrid, Madrid, Spain, 2004.
- [18] E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "Multi-sensor out-of-sequence data fusion for estimating the state of dynamic systems," in *Proc. IEEE Inform., Decision Control Conf.*, Feb. 2007, pp. 348–353.
- [19] Z. Jia and A. Balasuriya, "Visual target tracking for autonomous vehicles with the optimal oosm solution," in *Proc. 8th Int. Conf. Intell. Transport. Syst.*, 2005, pp. 113–118.